

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-036176

(43)Date of publication of application : 07.02.2003

(51)Int.Cl.

G06F 9/46

(21)Application number : 2001-223294 (71)Applicant : SONY CORP

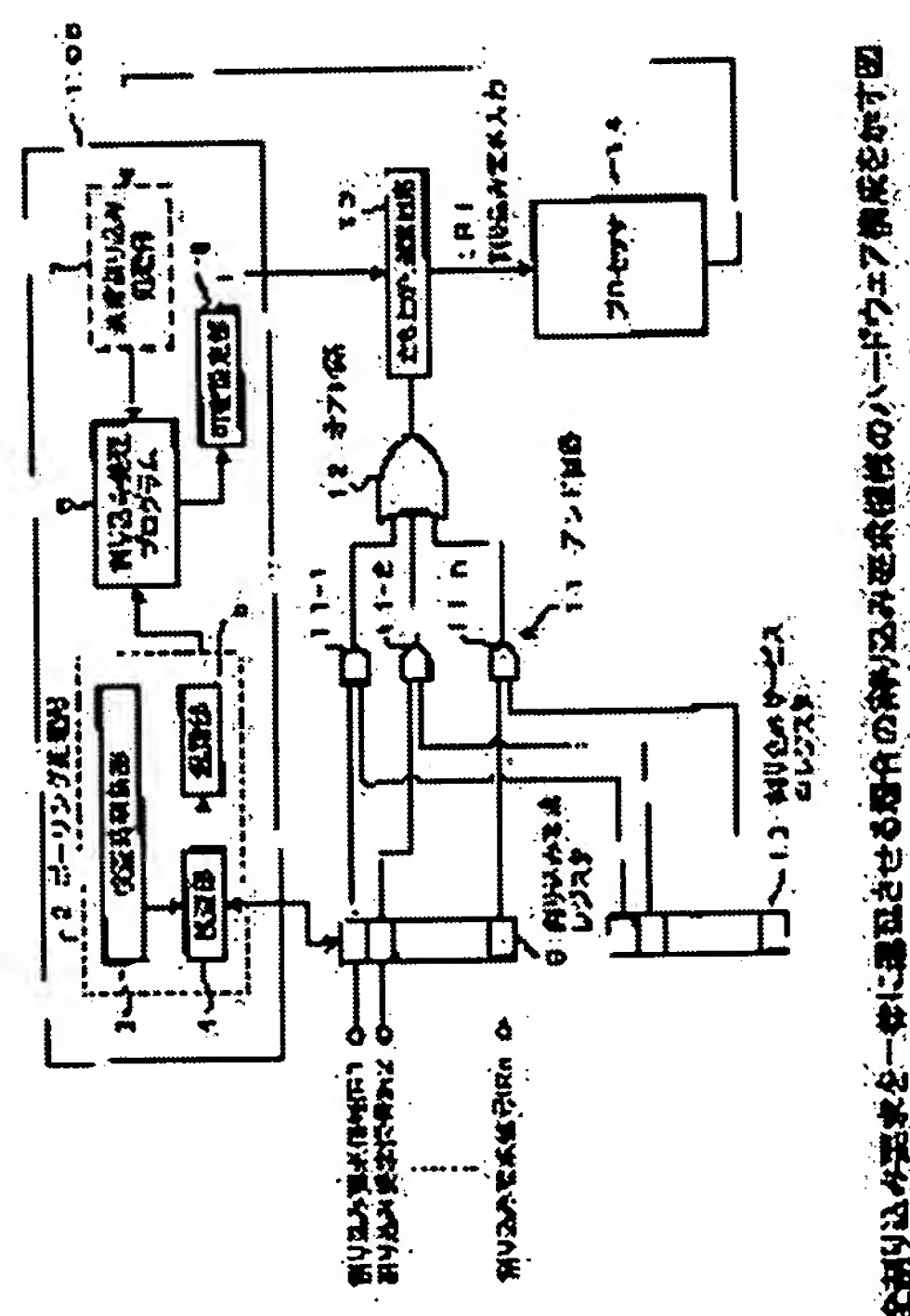
(22)Date of filing : 24.07.2001 (72)Inventor : TOGAWA ATSUSHI

(54) INTERRUPTION PROCESSING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an interruption processing system capable of avoiding performance deterioration also with respect to processing in which the flow of control changes at an unpredictable time such as interruption processing.

SOLUTION: The interruption processing system is provided with an interruption processing program 6 for executing the program of interruption processing in a computer system, a polling processing part 2 which inspects the occurrence of an interruption request by polling anywhere in an optional program and starts the program 6 when detecting the occurrence of the interruption request, an ordinary interruption processing part 7 for starting the interruption processing program by ordinary interruption processing based on the occurrence of the interruption request, a delay processing part which delays the ordinary interruption processing by the processing part 7 by a prescribed time.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than
the examiner's decision of rejection or
application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's
decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(51) Int.Cl. ⁷	識別記号	F I	テ-ミ-ト* (参考)
G 0 6 F 9/46	3 1 0	G 0 6 F 9/46	3 1 0 N 5 B 0 9 8
	3 1 1		3 1 1 A
	3 1 5		3 1 5 Z

審査請求 未請求 請求項の数 7 O L (全 11 頁)

(21) 出願番号	特願2001-223294 (P2001-223294)	(71) 出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22) 出願日	平成13年7月24日 (2001. 7. 24)	(72) 発明者	戸川 敦之 東京都品川区北品川6丁目7番35号 ソニー株式会社内
		(74) 代理人	100080883 弁理士 松隈 秀盛
		Fターム(参考)	5B098 AA02 BA01 BA04 BB05 BB16 EE06 FF02 FF04

(54) 【発明の名称】 割り込み処理システム

(57) 【要約】

【課題】 割り込み処理のように、予め予測できない時点に制御の流れが変化する処理に対しても、性能を低下させないようにすることができる割り込み処理システムを提供する。

【解決手段】 割り込み処理システムは、コンピュータシステムにおける割り込み処理のプログラムを実行する割り込み処理プログラム6と、任意のプログラム中の随所でポーリングによる割り込み要求発生の検査を行い、割り込み要求発生が検出されたときに割り込み処理プログラム6を起動させるポーリング処理部2と、割り込み要求発生に基づいて通常の割り込み処理により割り込み処理プログラムを起動させる通常割り込み処理部7と、通常割り込み処理部7による通常の割り込み処理を所定時間だけ遅延させる遅延処理部とを備えたものである。

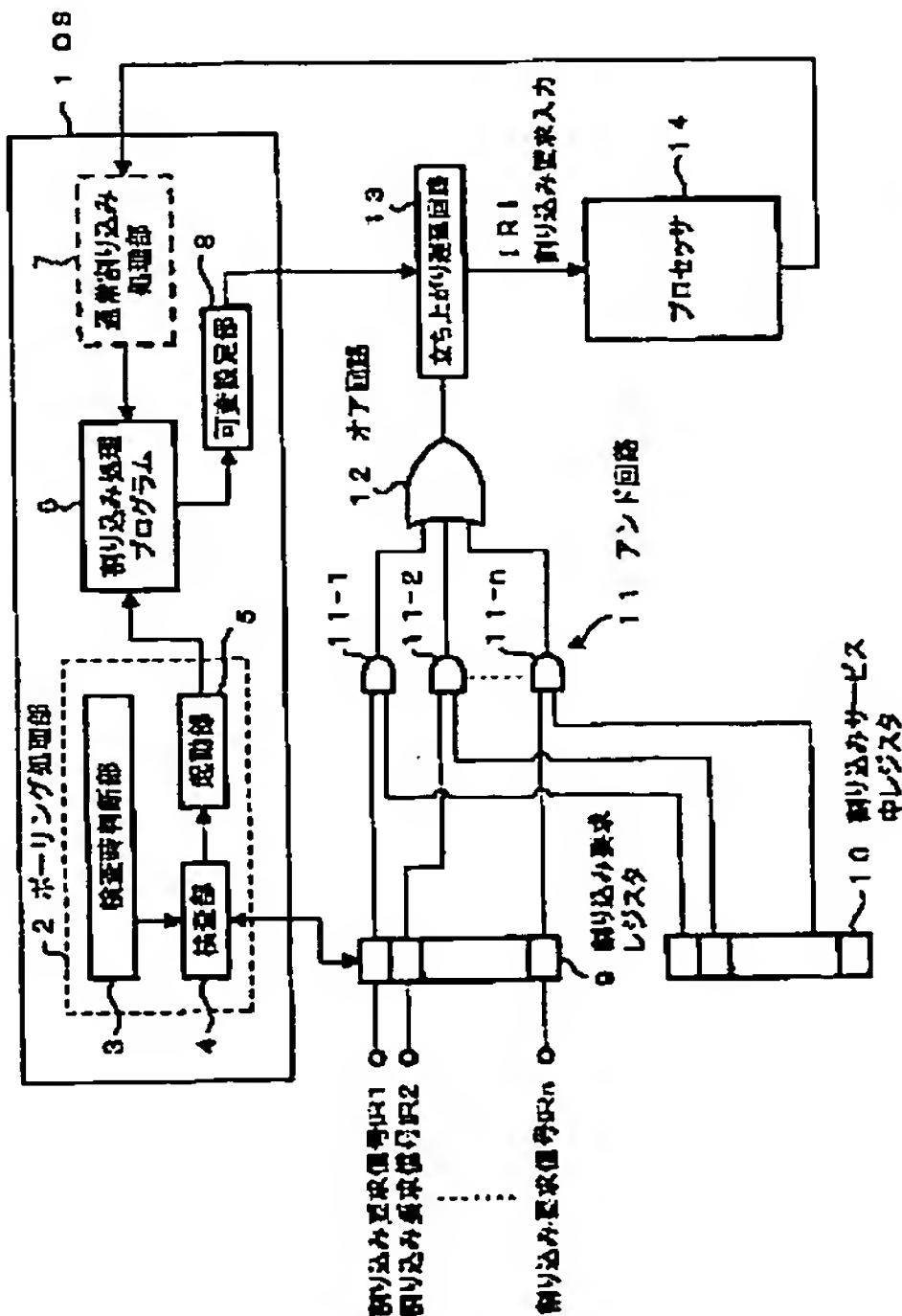


図1 全割り込み要求を一律に遅延させる場合の割り込み要求検出のハードウェア構成を示す図

【特許請求の範囲】

【請求項1】 コンピュータシステムにおける割り込み処理を行う割り込み処理システムにおいて、上記コンピュータシステムにおける割り込み処理のプログラムを実行する割り込み処理プログラム実行部と、上記コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生を検査を行い、割り込み要求発生が検出されたときに上記割り込み処理プログラムを起動させるポーリング処理部と、上記割り込み要求発生に基づいて通常の割り込み処理により上記割り込み処理プログラムを起動させる通常割り込み処理部と、上記通常割り込み処理部による通常の割り込み処理を所定時間だけ遅延させる遅延処理部と、を備えたことを特徴とする割り込み処理システム。

【請求項2】 請求項1記載の割り込み処理システムにおいて、上記ポーリング処理部は、上記コンピュータシステムのプログラムの実行の可能性の低い時点を判断する検査時判断部と、上記検査時判断部により判断された時点で割り込み要求発生の検査を行う検査部と、上記検査部により割り込み要求発生が検出されたときに上記割り込み処理プログラムを起動させる起動部と、を有することを特徴とする割り込み処理システム。

【請求項3】 請求項1記載の割り込み処理システムにおいて、上記遅延処理部による遅延時間を上記ポーリング処理部による検査時に応じて可変に設定する可変設定部を有することを特徴とする割り込み処理システム。

【請求項4】 請求項1記載の割り込み処理システムにおいて、上記遅延処理部は、複数の割り込み要求を一律に遅延させることを特徴とする割り込み処理システム。

【請求項5】 請求項1記載の割り込み処理システムにおいて、上記遅延処理部は、複数の割り込み要求を各割り込み発生源毎に遅延時間を変えて遅延させることを特徴とする割り込み処理システム。

【請求項6】 コンピュータシステムにおけるタイマ割り込み処理を行う割り込み処理システムにおいて、上記コンピュータシステムにおけるタイマ割り込み処理のプログラムを実行するタイマ割り込み処理プログラム実行部と、上記コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生の検査を行い、タイマ割り込み要求発生が検出されたときに上記タイマ割り込み処理プログラムを起動させるポーリング処理部と、上記タイマ割り込み要求発生に基づいて通常の割り込み

処理により上記タイマ割り込み処理プログラムを起動させるタイマ割り込み処理部と、上記タイマ割り込み処理部によるタイマ割り込み処理を所定時間だけ遅延させる遅延処理部と、を備えたことを特徴とする割り込み処理システム。

【請求項7】 請求項6記載の割り込み処理システムにおいて、上記遅延処理部は、複数のタイマ割り込み要求の処理の前後を更新することを特徴とする割り込み処理システム。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、例えば、オペレーティング・システム（OS）に適用して好適な割り込み処理システムに関するものである。

【0002】

【従来の技術】 一般に、コンピュータシステムは、周辺のハードウェアにおいてソフトウェアの介在が必要な事象が発生した際に、割り込み要求をプロセッサに送出することによって、その時点で実行されていたプログラムの実行を中断し、割り込み処理プログラムを起動する機構を有することが一般的であった。

【0003】 一方、近年のプロセッサは、以下の機構を採用することによって、その性能を飛躍的に高めている。第1に、多数のステージからなるパイプライン構造を設けるようにしていた。第2に、キャッシュメモリを設けるようにしていた。第3に、多数のレジスタを設けるようにしていた。第4に、分岐予測機構を設けるようにしていた。

【0004】 このような従来の手法では、割り込み処理要求の発生をプロセッサが検知し、これを受けて、オペレーティングシステムが現在のプロセッサの状態を保存した後に、割り込み処理プログラムを起動していた。そして、割り込み処理の完了時に、保存されていた状態を復元することによって、割り込みによって中断されていた処理を再開していた。

【0005】

【発明が解決しようとする課題】 上述した従来の機構は、割り込み処理のように、予め予測できない時点に制御の流れが変化する処理に対する性能が相対的に低くなるという不都合があった。

【0006】 さらに、いわゆるギガビットイーサネット（登録商標）などのように12 μ sec毎に割り込みが発生し得るより高速な通信機構では、大変高い頻度で割り込み要求が発生する。例えば、1Gbpsの通信速度で1500バイトのパケットを受信した場合、12マイクロ秒毎に1パケットが受信されることになる。パケット受信毎に割り込みを発生するという一般的なハードウェア構成を採用した場合、12マイクロ秒ごとに割り込み処理を行う必要が生じることになるため、このような

高速割り込みに対応し難いという不都合があった。

【0007】また、セットトップボックスなどのように、内部に多数の割り込み発生源を格納する必要があるシステムにおいても、割り込み処理に費やされる時間が全処理時間に占める割合が高くなる傾向があるという不都合があった。

【0008】仮に、割り込み発生間隔が予測可能であれば、オペレーティングシステムによるタイマデバイスのポーリングによって、この問題を解決することが可能である。(参考文献: Mohit Aron and Peter Druschel, "Soft Timer Efficient Microsecond Software Timer Support for Network Processing," ACM Transaction on Computer Systems, vol. 18, No. 3, August 2000.)

【0009】しかし、予め割り込み発生間隔が予測できない割り込み発生源では、このような手法を適用することができないという不都合があった。上述した従来の機構では、第1の多数のステージからなるパイプライン構造では、割り込み時の途中の処理の無駄となる。具体的には、一般的に、プロセッサは割り込み処理を開始または終了する前に、パイプライン中に残されている命令実行の途中結果が一掃されるまで待つか、あるいはこれらを捨てる必要がある。これによって、パイプライン中の各ステージの稼働率が低下し、結果として、プロセッサの性能が低下するという不都合があった。

【0010】また、第2のキャッシュメモリでは、割り込み時の処理復帰後の性能低下となる。具体的には、割り込み処理プログラムの実行によって、プロセッサのキャッシュメモリが、割り込みによって中断されたプログラムとは無関係な内容に書き換えられる。そのため、割り込み処理から復帰したとき、中断されていたプログラムの実行効率が低下するという不都合があった。

【0011】また、第3の多数のレジスタでは、プロセッサの状態を記憶するための処理が多くなる。具体的には、今日のマイクロプロセッサは多数のレジスタを備えている。そのため、これらを保存または復元する処理に多くの時間を費やすという不都合があった。

【0012】また、第4の分岐予測機構では、割り込み直後の予測が外れやすくなる。具体的には、分岐予測機構を備えたプロセッサの場合、割り込み処理プログラムの実行によって分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まるため、性能が低下するという不都合があった。

【0013】そこで、本発明は、かかる点に鑑みてなされたものであり、割り込み処理のように、予め予測できない時点に制御の流れが変化する処理に対しても、性能を低下させないようにすることができる割り込み処理シ

ステムを提供することを課題とする。

【0014】

【課題を解決するための手段】本発明の割り込み処理システムは、コンピュータシステムにおける割り込み処理を行う割り込み処理システムにおいて、コンピュータシステムにおける割り込み処理のプログラムを実行する割り込み処理プログラム実行部と、コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生を検査を行い、割り込み要求発生が検出されたときに割り込み処理プログラムを起動させるポーリング処理部と、割り込み要求発生に基づいて通常の割り込み処理により割り込み処理プログラムを起動させる通常割り込み処理部と、通常割り込み処理部による通常の割り込み処理を所定時間だけ遅延させる遅延処理部とを備えたものである。

【0015】従って本発明によれば、以下の作用をする。オペレーティングシステムが所定の処理を行うときをポーリング処理部の検査時判断部が判断して、ポーリング処理部の検査部が割り込み要求の検査を行う。このような、ポーリング処理により処理のオーバーヘッドを削減する。

【0016】また、ポーリング処理部に加えて、割り込み要求が発生してからプロセッサに割り込み要求入力に到達するまでの時間を意図的に遅らせる遅延処理部を併用する。

【0017】これによって、ポーリング処理部によるオーバーヘッドの削減効果を維持しつつ、遅延処理部により、ある一定時間以上、割り込み要求発生の検知が遅れないことを保証する。

【0018】ポーリング処理部の検査時判断部による判断時点で、検査部が割り込み要求の検査を行う。この検査部による検査が十分高い頻度で行われなかった際に、すなわち、遅延された割り込み要求入力がプロセッサに到達されたときに、通常割り込み処理部により割り込み処理プログラムが起動される。この遅延時間に、通常のシステムにおける割り込み応答時間を加えた時間が、最大割り込み応答時間となる。

【0019】このような全割り込み要求を遅延させる遅延処理部を用いることにより、割り込み要求信号を遅延させることができるので、システムの最大割り込み応答時間を保証しつつ、ポーリング処理部による割り込み処理の起動の確率を高くすることができる。

【0020】また、本発明の割り込み処理システムは、コンピュータシステムにおけるタイマ割り込み処理を行う割り込み処理システムにおいて、コンピュータシステムにおけるタイマ割り込み処理のプログラムを実行するタイマ割り込み処理プログラム実行部と、コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生を検査を行い、タイマ割り込み要求発生が検出されたときにタイマ割り込み処理プログラム

を起動させるポーリング処理部と、タイマ割り込み要求発生に基づいて通常の割り込み処理によりタイマ割り込み処理プログラムを起動させるタイマ割り込み処理部と、タイマ割り込み処理部によるタイマ割り込み処理を所定時間だけ遅延させる遅延処理部とを備えたものである。

【0021】従って本発明によれば、以下の作用をする。遅延処理部の設定タイマリストには、設定後、いまだ設定時刻に達していないタイマごとに、タイマ待ち情報が用意されている。これらタイマ待ち情報は、双方向リンクリストを構成する。

【0022】また、遅延処理部のタイマ待ち情報は、タイマ設定時刻、許容遅延時間、手続きポインタを有している。

【0023】遅延処理部の設定タイマリストに対して、タイマ待ち情報は、タイマ設定時刻が短いものから順に並べられている。手続きポインタは、設定時間経過後に呼び出すべき手続きを指示するポインタである。

【0024】タイマ待ち情報のタイマ設定時刻の経過後に、まだ設定時間が経過していないタイマ設定時刻のタイマ待ち情報の手続きを手続きポインタが呼び出す指示をする。

【0025】これによって、ポーリング処理部によって設定タイマリストに対してタイマ待ち情報のタイマの時間切れ発生の検査をすることによりオーバーヘッド削減効果を得つつ、ある一定時間以上、設定タイマリストに対してタイマ待ち情報のタイマ検知が遅れないことを保証する。

【0026】すなわち、タイマ待ち情報のタイマ設定時刻に対する許容遅延時間により、タイマ応答時間の最大値を超えない範囲で、タイマ設定時刻よりも遅い時間にタイマ割り込み処理部によりタイマ割り込み処理プログラムに対して割り込みを発生させるように設定タイマリストを設定する。

【0027】これによって、タイマ割り込み処理部によりタイマ割り込み処理プログラムに対して割り込みが発生する前にポーリング処理部によって設定タイマリストに対してタイマ待ち情報のタイマ設定時刻となったことが検知される可能性を高める。

【0028】

【発明の実施の形態】本発明の実施の形態を以下に説明する。本実施の形態の割り込み処理システムは、オペレーティングシステムによる割り込み要求信号のポーリング機構と、割り込み要求信号に対するプロセッサの割り込み応答時間を意図的に遅らせる機構とを組み合わせることによって、割り込み処理によって発生するオーバーヘッドを削減するものである。

【0029】図1は、本実施の形態に適用される全割り込み要求を一律に遅延させる場合の割り込み要求機構のハードウェア構成を示す図である。上述した従来の機構

で、第1の多数のステージからなるパイプライン構造による、プロセッサの性能が低下するという不都合、および第3の多数のレジスタによる、これらを保存または復元する処理に多くの時間を費やすという不都合については、予め予測できない時点に割り込み処理が開始されるために発生する問題である。

【0030】したがって、図1において示すように、オペレーティングシステム（OS）1内で実行されるプログラム中に、検査部4により割り込み要求信号IR1、IR2・・・IRnが入力される割り込み要求レジスタ9における割り込み要求の発生を検査し、必要に応じて起動部5により割り込み処理プログラム6を呼び出すポーリング処理部2を埋め込むことによって、上述した問題を解決することができる。

【0031】ここで、問題となるのは、どのプログラムの、どの位置にこのコードを埋め込むかということである。このポーリング処理が有効であるためには、十分高い頻度で実行される箇所に、割り込み要求検査コードを挿入する必要がある。

【0032】さらに、アプリケーションプログラムのキャッシュメモリヒット率や分岐予測効率を悪化させないためには、あるプログラム（例えば、アプリケーションプログラムとは限らず、オペレーティングシステムソフトウェアであっても良い。）から他のアプリケーションプログラムへ制御が移行する直前か、または、引き続きアプリケーションプログラムが実行される可能性が低い箇所で検査を行うことが望ましい。

【0033】このような位置で検査を行うことによって、上述した問題の内、第2のキャッシュメモリによる、割り込み処理から復帰したとき、中断されていたプログラムの実行効率が低下するという不都合、および第4の分岐予測機構による、割り込み処理プログラムの実行によって分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まるため、性能が低下するという不都合を解決することができる。

【0034】そこで、オペレーティングシステムが以下の処理を行うときをポーリング処理部2の検査時判断部3が判断して、ポーリング処理部2の検査部4が割り込み要求の検査を行うようにする。

【0035】第1に、システムコール処理の終了直前に行う。ここで、システムコールとは、オペレーティングシステムに内蔵されている入出力ルーチンなどを、ユーザやアプリケーションプログラムが使うことを指している。

【0036】第2に、例外処理の終了直前、例えば、メモリ保護例外処理の終了直前に行う。第3に、CPUがなにもプログラムを実行していないアイドル状態にある時点に行う。

【0037】上述した第1、第2、および第3の場合に

は、コンパイラがレジスタの割付をするので、復元すべきレジスタ数が半分程度で済むことになる。

【0038】これらの処理は、本実施の形態の目的に十分なほど頻繁に実行されることが上述した参考文献で報告されている。このような、ポーリング処理により上述した処理のオーバーヘッドを削減することができる。

【0039】ところが、割り込み応答時間の最大値を保証する必要がある場合がしばしばある。このような場合、オペレーティングシステム（OS）1によるポーリング処理部2だけでは不十分である。また、最大値を保証する必要はないものの、応答時間の増大が性能の劣化につながる場合も多い。

【0040】そこで、本実施の形態では、上述したポーリング処理部2に加えて、割り込み要求が発生してからプロセッサ14に割り込み要求入力IRiが到達するまでの時間を意図的に遅らせる遅延機能を有する割り込み要求機構を併用することを提案する。

【0041】これによって、ポーリング処理部2によるオーバーヘッドの削減効果を維持しつつ、ある一定時間以上、割り込み要求発生を検知が遅れないことを保証することができる。

【0042】上述した検査時判断部3による第1、第2および第3の判断時点で、検査部4が割り込み要求の検査を行う。この検査部4による検査が十分高い頻度で行われなかった際に、すなわち、遅延された割り込み要求入力IRiがプロセッサ14に到達されたときに、通常割り込み処理部7により割り込み処理プログラム6が起動されることになる。

【0043】以下に、遅延機能を有する割り込み要求機構について説明する。割り込み要求信号IR1、IR2、・・・IRnは、ある割り込み発生源が割り込み処理を要求しているとき、ハイレベルH（または1）になる信号である。

【0044】割り込み要求レジスタ9は、現在発生している割り込み要求を、オペレーティングシステム（OS）1内のポーリング処理部2の検査部4が読み出すことが可能なレジスタである。

【0045】割り込みサービス中レジスタ10は、アンド回路11およびオア回路12を介して、プロセッサ14に割り込み要求入力IRiを伝達するためのレジスタであり、オペレーティングシステム（OS）1内のポーリング処理部2の検査部4が読み出すことが可能なレジスタである。

【0046】立ち上がり遅延回路13は、入力信号がローレベルL（または0）からハイレベルH（または1）に変化してから、一定期間経過後に出力をハイレベルH（または1）に変化させ、入力信号がハイレベルH（または1）からローレベルL（または0）に変化したときには、即座に出力をローレベルL（または0）に変化させる回路である。

【0047】この遅延時間に、通常のシステムにおける割り込み応答時間を加えた時間が、最大割り込み応答時間となる。具体的には、例えば、システム仕様上で保証したい最大応答時間が500μsecのとき、オペレーティングシステム（OS）1が割り込み処理に要する時間10μsecを引いた490μsecを遅延時間として設定するようにする。

【0048】また、可変設定部8は、立ち上がり遅延回路13に対してソフトウェアにより遅延時間を変化可能に設定するものである。ここで、割り込み要求時間に対して厳しい要求を課している割り込み発生源を備えたシステムも存在している。このような場合に対応するため、遅延時間を0にできるよう構成することも可能である。

【0049】このような全割り込み要求を一律に遅延させる場合の割り込み要求機構を用いることにより、全ての割り込み要求信号の論理和を一律に遅延させることができるので、システムの最大割り込み応答時間を保証しつつ、ポーリング処理部2による割り込み処理の起動の確率を高くすることができる。

【0050】図2は、各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構のハードウェア構成を示す図である。ここでは、図1と同一の構成は同一の符号を付して、その説明を省略し、異なる構成のみを説明する。

【0051】割り込みマスクレジスタ15は、アンド回路11およびオア回路12を介して、プロセッサ14への割り込み要求入力IRiの伝達を制限するためのレジスタであり、オペレーティングシステム（OS）1内のポーリング処理部2の検査部4が読み出しまたは書き込みが可能なレジスタである。

【0052】立ち上がり遅延回路13は、各割り込み発生源ごとに遅延時間を変えて設定され、入力信号がローレベルL（または0）からハイレベルH（または1）に変化してから、一定期間経過後に出力をハイレベルH（または1）に変化させ、入力信号がハイレベルH（または1）からローレベルL（または0）に変化したときには、即座に出力をローレベルL（または0）に変化させる回路である。

【0053】図2においては、立ち上がり遅延回路13-1、13-2・・・13-nにより各割り込み発生源毎に異なる遅延時間による遅延を行った後に、オア回路12によりそれらの論理和をとるようにしている。

【0054】このような各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構を用いることにより、各割り込み発生源毎に最適な遅延時間を選択して遅延させることができるので、システムの最大割り込み応答時間および各割り込み発生源毎に最適な遅延時間を保証しつつ、ポーリング処理部2による割り込み処理の起動の確率を高くすることができる。

【0055】図1に示した全割り込み要求を一律に遅延させる場合の割り込み要求機構は図2に示した各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構に比べて、ハードウェアの規模を小さくすることができるという効果がある。

【0056】これに対して、図2に示した各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構は、各割り込み発生源毎に最適な遅延時間を設定することができるという効果を奏する。

【0057】また、これに限らず、図1に示した全割り込み要求を一律に遅延させる場合の割り込み要求機構と、図2に示した各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構とを組み合わせ適用することも可能である。

【0058】すなわち、一部の割り込み発生源については図1に示すようにオア回路12によりそれらの論理和の後に立ち上がり遅延回路13-1、13-2・・・13-nにより遅延させるようにし、他の割り込み発生源については図2に示すように個別に立ち上がり遅延回路13-1、13-2・・・13-nにより遅延を行ってからオア回路12によりそれらの論理和を取るように構成しても良い。

【0059】このように本実施の形態によれば、以下のような性能低下を削減する効果が得られる。まず、割り込み処理の開始または終了時に、多数のレジスタの状態を保存または復元する処理に要する多くの時間の消費を低減することができる。

【0060】また、割り込み処理の開始または終了する前に、多数のステージからなるパイプライン中に残されている命令実行の途中結果が一掃されるまで待つか、あるいはこれらを捨てることによる、パイプライン中の各ステージの稼働率の低下、結果として、プロセッサの性能低下を低減することができる。

【0061】また、割り込み処理プログラムの実行によるキャッシュメモリが、割り込みによって中断されたプログラムとは無関係な内容に書き換えられることによる、割り込み処理から復帰したときに中断されていたプログラムの実行効率の低下を低減することができる。

【0062】また、割り込み処理プログラムの実行による分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まることによる、性能の低下を低減することができる。

【0063】次に、本実施の形態の他の割り込み処理システムについて説明する。本実施の形態の他の割り込み処理システムは、オペレーティングシステムによるタイマの時間切れ発生の検査をするポーリング処理部と、タイマ割り込みの発生を意図的に遅らせるデータ構造とを組み合わせることによって、タイマ割り込み処理によって発生するオーバーヘッドを削減するものである。

【0064】図3は、タイマ割り込みを遅延させる場合

の割り込み要求のデータ構造を示す図である。図3において、ポーリング処理部2は、上述した図1および図2に示したものと同様であるので、その説明を省略する。設定タイマリスト31には、設定後、いまだ設定時刻に達していないタイマごとに、タイマ待ち情報32-1、32-2、32-3・・・が用意されている。これらタイマ待ち情報32-1、32-2、32-3・・・は、双方向リンクリストを構成している。

【0065】また、タイマ待ち情報32-1は、タイマ設定時刻32-1-1、許容遅延時間32-1-2、手続きポインタ32-1-3を有している。また、タイマ待ち情報32-2は、タイマ設定時刻32-2-1、許容遅延時間32-2-2、手続きポインタ32-2-3を有している。また、タイマ待ち情報32-3は、タイマ設定時刻32-3-1、許容遅延時間32-3-2、手続きポインタ32-3-3を有している。

【0066】設定タイマリスト31に対して、タイマ待ち情報32-1、32-2、32-3・・・は、タイマ設定時刻32-1-1、タイマ設定時刻32-2-1、タイマ設定時刻32-3-1・・・が短いものから順に並べられている。手続きポインタ32-1-3、手続きポインタ32-2-3、手続きポインタ32-3-3は、設定時間経過後に呼び出すべき手続きを指示するポインタである。

【0067】具体的には、タイマ待ち情報32-1のタイマ設定時刻32-1-1経過後に、まだ設定時間が経過していないタイマ設定時刻32-2-1のタイマ待ち情報32-2の手続きを手続きポインタ32-2-3が呼び出す指示をする。このとき、タイマ待ち情報32-1に替えてタイマ待ち情報32-2を最前列に並べ替える。

【0068】また、タイマ待ち情報32-2のタイマ設定時刻32-2-1経過後に、まだ設定時間が経過していないタイマ設定時刻32-3-1のタイマ待ち情報32-3の手続きを手続きポインタ32-3-3が呼び出す指示をする。このとき、タイマ待ち情報32-2に替えてタイマ待ち情報32-3を最前列に並べ替える。

【0069】これによって、ポーリング処理部2によって設定タイマリスト31に対してタイマ待ち情報32-1、32-2、32-3・・・のタイマの時間切れ発生の検査をすることによりオーバーヘッド削減効果を得つつ、ある一定時間以上、設定タイマリスト31に対してタイマ待ち情報32-1、32-2、32-3・・・のタイマ検知が遅れないことを保証することができる。

【0070】すなわち、タイマ待ち情報32-1、32-2、32-3・・・のタイマ設定時刻32-1-1、32-2-1、32-3-1・・・に対する許容遅延時間32-1-2、32-2-2、32-3-2・・・により、タイマ応答時間の最大値を超えない範囲で、タイマ設定時刻32-1-1、32-2-1、32-3-1

・・・よりも遅い時間にタイマ割り込み処理部34によりタイマ割り込み処理プログラム33に対して割り込みを発生させるように設定タイマリスト31を設定する。

【0071】これによって、タイマ割り込み処理部34によりタイマ割り込み処理プログラム33に対して割り込みが発生する前にポーリング処理部2によって設定タイマリスト31に対してタイマ待ち情報32-1、32-2、32-3・・・のタイマ設定時刻32-1-1、32-2-1、32-3-1・・・となったことが検知される可能性を高めることができる。

【0072】図4は、タイマ登録処理の手順を示すフローチャートである。図4のタイマ登録処理は、オペレーティングシステム(OS)1内のアプリケーションプログラムが設定タイマリスト31に対するタイマ待ち情報32-1、32-2、32-3・・・のように新しくタイマ登録するときの動作を示すものである。

【0073】図4において、ステップS1で、現在時刻がタイマ設定時刻より小さいか否かを判断する。ステップS1で現在時刻がタイマ設定時刻より小さいときは、ステップS2へ進み、ステップS2で、タイマ待ち情報のための領域を割り当てる。ステップS3で、タイマ待ち情報を初期化する。

【0074】ステップS4で、設定タイマリストに挿入する。ステップS5で、先頭に挿入されたか否かを判断する。ステップS5で先頭に挿入されたときは、ステップS6へ進んで、ステップS6で、設定時刻+許容遅延時間に割り込みを発生させるようにタイマハードウェアを設定して、タイマ登録処理を完了する。

【0075】ステップS1で現在時刻がタイマ設定時刻より小さいときは、ステップS7へ進み、ステップS7で、設定時間経過後に呼び出すべき手続きを呼び出して、タイマ登録処理を完了する。

【0076】図5は、ポーリング処理の手順を示すフローチャートである。図5のポーリング処理は、オペレーティングシステム(OS)1内のポーリング処理部2が設定タイマリスト31に対してタイマの時間切れ発生の検査をするときの動作を示すものである。

【0077】図5において、ステップS11で、設定タイマリストは空か否かを判断する。ステップS11で設定タイマリストが空でないときは、ステップS12へ進んで、ステップS12で、先頭タイマ待ち情報のタイマ設定時刻に達しているか否かを判断する。

【0078】ステップS12で先頭タイマ待ち情報のタイマ設定時刻に達しているときは、ステップS13へ進んで、ステップS13で、先頭タイマ待ち情報の除去を行う。

【0079】ステップS14で、設定時間経過後に呼び出すべき手続きを呼び出して、ステップS11へ戻って、ステップS11～ステップS14までの判断および処理を繰り返す。そして、ステップS11で設定タイマ

リストが空のときは、ポーリング処理を完了する。

【0080】図6は、タイマ割り込み処理の手順を示すフローチャートである。図6のタイマ割り込み処理は、オペレーティングシステム(OS)1内のタイマ割り込み処理部34が設定タイマリスト31に対してタイマ割り込み処理を行うときの動作を示すものである。

【0081】図6において、ステップS21で、先頭タイマ待ち情報の除去を行う。ステップS22で、設定時間経過後に呼び出すべき手続きを呼び出す。ステップS23で、設定タイマリストは空か否かを判断する。ステップS23で設定タイマリストが空でないときは、ステップS24へ進み、ステップS24で、先頭タイマ待ち情報のタイマ設定時刻に達しているか否かを判断する。

【0082】ステップS24で先頭タイマ待ち情報のタイマ設定時刻に達しているときは、ステップS21へ戻り、ステップS21～ステップS24までの判断および処理を繰り返す。そして、ステップS23で設定タイマリストが空のとき、および、ステップS24で先頭タイマ待ち情報のタイマ設定時刻に達していないときは、タイマ割り込み処理を完了する。

【0083】図7は、先頭タイマ待ち情報の除去を示すフローチャートである。図7の先頭タイマ待ち情報の除去は、オペレーティングシステム(OS)1内のタイマ割り込み処理部34が設定タイマリスト31に対して先頭タイマ待ち情報の除去を行うときの動作のサブルーチンを示すものである。このサブルーチンは、図5に示したステップS13および図6に示したステップS21に対応するものである。

【0084】ステップS31で、先頭のタイマ待ち情報をリストから除去する。ステップS32で、設定タイマリストは空か否かを判断する。ステップS32で設定タイマリストが空でないときは、ステップS33へ進み、ステップS33で、先頭要素の設定時刻+許容遅延時間に割り込みを発生させるようにタイマハードウェアを設定して、先頭タイマ待ち情報の除去を完了する。ステップS32で設定タイマリストが空のときは、ステップS34へ進み、ステップS34で、割り込みを発生させないようにタイマハードウェアを設定する。

【0085】このように本実施の形態によれば、特別なハードウェアを必要とすることなく、タイマ応答時間の最大値を限定し、かつ、以下のような性能低下を削減する効果が得られる。

【0086】まず、タイマ割り込み処理の開始または終了時に、多数のレジスタの状態を保存または復元する処理に要する多くの時間の消費を低減することができる。

【0087】また、タイマ割り込み処理の開始または終了する前に、多数のステージからなるパイプライン中に残されている命令実行の途中結果が一掃されるまで待つか、あるいはこれらを捨てることによる、パイプライン中の各ステージの稼働率の低下、結果として、プロセッ

サの性能低下を低減することができる。

【0088】また、タイマ割り込み処理プログラムの実行によるキャッシュメモリが、割り込みによって中断されたプログラムとは無関係な内容に書き換えられることによる、割り込み処理から復帰したときに中断されていたプログラムの実行効率の低下を低減することができる。

【0089】また、タイマ割り込み処理プログラムの実行による分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まることによる、性能の低下を低減することができる。

【0090】

【発明の効果】この発明の割り込み処理システムは、コンピュータシステムにおける割り込み処理を行う割り込み処理システムにおいて、コンピュータシステムにおける割り込み処理のプログラムを実行する割り込み処理プログラム実行部と、コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生を検査を行い、割り込み要求発生が検出されたときに割り込み処理プログラムを起動させるポーリング処理部と、割り込み要求発生に基づいて通常の割り込み処理により割り込み処理プログラムを起動させる通常割り込み処理部と、通常割り込み処理部による通常の割り込み処理を所定時間だけ遅延させる遅延処理部とを備えたので、割り込み処理の開始または終了時に、多数のレジスタの状態を保存または復元する処理に要する多くの時間の消費を低減することができる。また、割り込み処理の開始または終了する前に、多数のステージからなるパイプライン中に残されている命令実行の途中結果が一掃されるまで待つか、あるいはこれらを捨てることによる、パイプライン中の各ステージの稼働率の低下、結果として、プロセッサの性能低下を低減することができる。また、割り込み処理プログラムの実行によるキャッシュメモリが、割り込みによって中断されたプログラムとは無関係な内容に書き換えられることによる、割り込み処理から復帰したときに中断されていたプログラムの実行効率の低下を低減することができる。また、割り込み処理プログラムの実行による分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まることによる、性能の低下を低減することができるという効果を奏する。

【0091】また、この発明の割り込み処理システムは、上述において、ポーリング処理部は、コンピュータシステムのプログラムの実行の可能性の低い時点と判断する検査時判断部と、検査時判断部により判断された時点で割り込み要求発生を検査を行う検査部と、検査部により割り込み要求発生が検出されたときに割り込み処理プログラムを起動させる起動部とを有するので、ポーリング処理により処理のオーバーヘッドを削減することができるという効果を奏する。

【0092】また、この発明の割り込み処理システムは、上述において、遅延処理部による遅延時間をポーリング処理部による検査時に応じて可変に設定する可変設定部を有するので、遅延処理部に対してソフトウェアにより遅延時間を変化可能に設定することができるという効果を奏する。

【0093】また、この発明の割り込み処理システムは、上述において、遅延処理部は、複数の割り込み要求を一律に遅延させるので、全ての割り込み要求信号の論理和を一律に遅延させることができるので、システムの最大割り込み応答時間を保証しつつ、ポーリング処理部による割り込み処理の起動の確率を高くすることができるという効果を奏する。

【0094】また、この発明の割り込み処理システムは、上述において、遅延処理部は、複数の割り込み要求を各割り込み発生源毎に遅延時間を変えて遅延させるので、各割り込み発生源毎に最適な遅延時間を設定することができるという効果を奏する。

【0095】また、この発明の割り込み処理システムは、コンピュータシステムにおけるタイマ割り込み処理を行う割り込み処理システムにおいて、コンピュータシステムにおけるタイマ割り込み処理のプログラムを実行するタイマ割り込み処理プログラム実行部と、コンピュータシステムの任意のプログラム中の随所でポーリングによる割り込み要求発生を検査を行い、タイマ割り込み要求発生が検出されたときにタイマ割り込み処理プログラムを起動させるポーリング処理部と、タイマ割り込み要求発生に基づいて通常の割り込み処理によりタイマ割り込み処理プログラムを起動させるタイマ割り込み処理部と、タイマ割り込み処理部によるタイマ割り込み処理を所定時間だけ遅延させる遅延処理部とを備えたので、特別なハードウェアを必要とすることなく、タイマ応答時間の最大値を限定し、かつ、以下のような性能低下を削減する効果が得られる。まず、タイマ割り込み処理の開始または終了時に、多数のレジスタの状態を保存または復元する処理に要する多くの時間の消費を低減することができる。また、タイマ割り込み処理の開始または終了する前に、多数のステージからなるパイプライン中に残されている命令実行の途中結果が一掃されるまで待つか、あるいはこれらを捨てることによる、パイプライン中の各ステージの稼働率の低下、結果として、プロセッサの性能低下を低減することができる。また、タイマ割り込み処理プログラムの実行によるキャッシュメモリが、割り込みによって中断されたプログラムとは無関係な内容に書き換えられることによる、割り込み処理から復帰したときに中断されていたプログラムの実行効率の低下を低減することができる。また、タイマ割り込み処理プログラムの実行による分岐予測バッファの内容が更新され、割り込み処理からの復帰後に分岐予測ミスの発生率が高まることによる、性能の低下を低減することが

ある。

【図6】タイマ割り込み処理の手順を示すフローチャートである。

【図7】先頭タイマ待ち情報の除去を示すフローチャートである。

１……オペレーティングシステム（OS）、２……ポーリング処理部、３……検査時判断部、４……検査部、５……起動部、６……割り込み処理プログラム、７……通常割り込み処理部、８……可変設定部、９……割り込み要求レジスタ、１０……割り込みサービス中レジスタ、１１……アンド回路、１２……オア回路、１３……立ち上がり遅延回路、１４……プロセッサ、１５……割り込みマスクレジスタ、３１……設定タイマリスト、３２－１……タイマ待ち情報、３２－１－１……タイマ設定時刻、３２－１－２……許容遅延時間、３２－１－３……手続きポインタ、３２－２……タイマ待ち情報、３２－２－１……タイマ設定時刻、３２－２－２……許容遅延時間、３２－２－３……手続きポインタ、３２－３……タイマ待ち情報、３２－３－１……タイマ設定時刻、３２－３－２……許容遅延時間、３２－３－３……手続きポインタ、３３……タイマ割り込み処理プログラム、３４……タイマ割り込み処理部、

【図 1】本実施の形態に適用される全割り込み要求を一
 律に遅延させる場合の割り込み要求機構のハードウェア
 構成を示す図である。

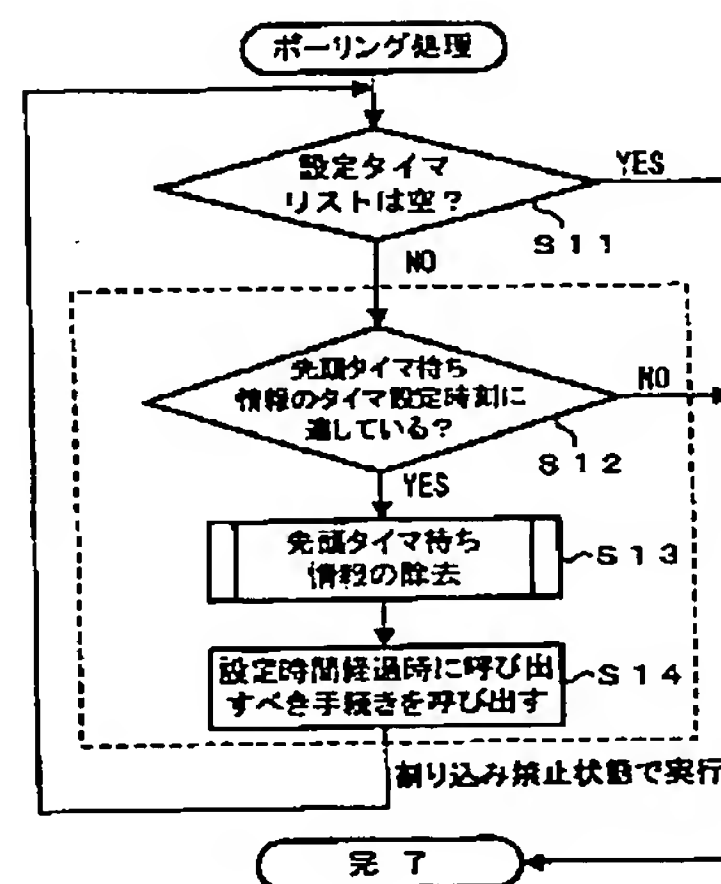
【図2】各割り込み要求ごとに遅延時間を変える場合の割り込み要求機構のハードウェア構成を示す図である。

【図3】タイマ割り込みを遅延させる場合の割り込み要求のデータ構造を示す図である。

【図4】タイマ登録処理の手順を示すフローチャートである。

【図5】ポーリング処理の手順を示すフローチャートで

【图 5】



ポーリング処理の手順を示すフローチャート

[illegible]

各割り込み発生源ごとに遅延時間を変える場合の割り込み要求機構のハードウェア構成を示す図

Figure 1 is a block diagram of the timer control system. The diagram shows a sequence of operations starting from a '2 ポーリング処理部' (Polling Processing Unit) which contains a '3 検査時判断部' (Check Time Judgment Unit) and a '4 検査部' (Check Unit). The '3 検査時判断部' outputs to the '4 検査部', which then outputs to a '5 起動部' (Start Unit). The '5 起動部' outputs to a '33 タイマ割り込み処理プログラム' (Timer Interrupt Processing Program). This program outputs to a '34 タイマ割り込み処理部' (Timer Interrupt Processing Unit). The '34' unit outputs to a '1 OS' (Operating System). The '1 OS' outputs to a '31 設定タイマリスト' (Setting Timer List). The '31' list outputs to a '32-1 タイマ待ち情報' (Timer Waiting Information) block, which contains '32-1-1 タイマ設定時刻' (Timer Setting Time), '32-1-2 許容遅延時間' (Tolerable Delay Time), and '32-1-3 手続きポインタ' (Procedure Pointer). This block outputs to a '32-2 タイマ待ち情報' block, which contains '32-2-1 タイマ設定時刻', '32-2-2 許容遅延時間', and '32-2-3 手続きポインタ'. This block outputs to a '32-3 タイマ待ち情報' block, which contains '32-3-1 タイマ設定時刻', '32-3-2 許容遅延時間', and '32-3-3 手続きポインタ'. The '32-3' block outputs to a dashed line indicating further processing.

タイマ割り込みを遅延させる場合の割り込み要求のデータ構造を示す図

```

    graph TD
      Start([スタート]) --> S1[S1: 先頭タイマ待ち  
情報の除去]
      S1 --> S2[S2: 設定時間経過時に呼び出  
すべき手続を呼び出す]
      S2 --> D1{D1: 設定タイ  
マリストは空?}
      D1 -- YES --> S3[S3: 先頭タイマ待ち  
情報のタイマ設定時刻に  
達している?]
      D1 -- NO --> S3
      S3 -- YES --> End([完了])
      S3 -- NO --> S1
  
```

タイマ割り込み処理の手順を示すフローチャート

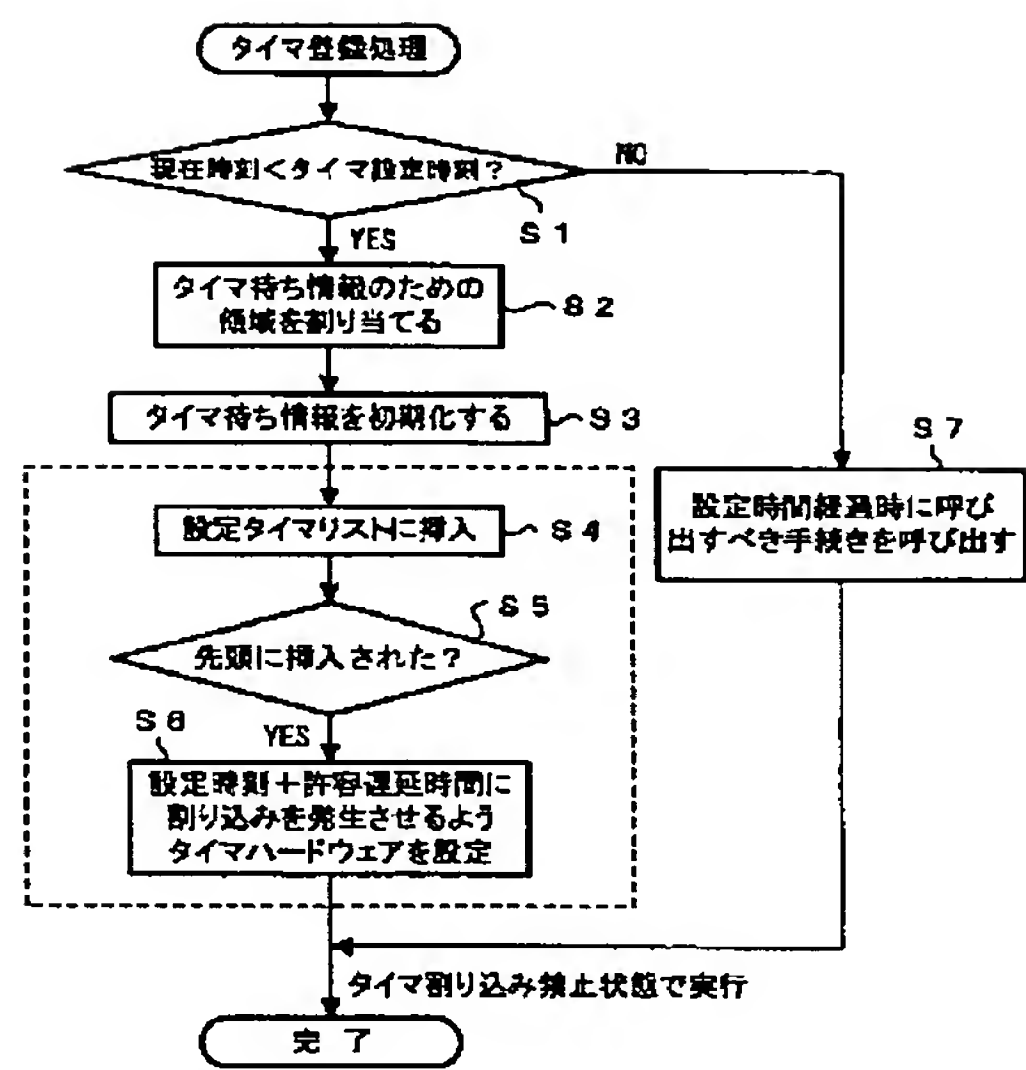
```

graph TD
    S30([先頭タイマ待ち情報の除去]) --> S31[先頭のタイマ待ち情報をリストから除去 S31]
    S31 --> S32{設定タイマリストは空? S32}
    S32 -- YES --> S34[割り込みを発生させないようタイマハードウェアを設定 S34]
    S32 -- NO --> S33[先頭優先の設定時刻+許容遅延時間に割り込みを発生させるようタイマハードウェアを設定 S33]
    S34 --> S35([完了])
    S33 --> S35
    
```

タイマ割り込み禁止状態で実行

先頭タイマ待ち情報の除去を示すフローチャート

【図4】



タイマ登録処理の手順を示す示すフローチャート